DISTANT: Distantly Supervised Entity Span Detection and Classification

Ken Yano[1], Makoto Miwa[1,2], and Sophia Ananiadou [1,3]

[1]Artificial Intelligence Research Center (AIRC)
[1]National Institute of Advanced Industrial Science and Technology (AIST), Japan
[2]Toyota Technological Institute, Japan
[3]Department of Computer Science, National Centre for Text Mining
[3]University of Manchester, United Kingdom
yano.ken@aist.go.jp, makoto-miwa@toyota-ti.ac.jp, sophia.ananiadou@manchester.ac.uk

Abstract

We propose a distantly supervised pipeline NER which executes entity span detection and entity classification in sequence named DISTANT (**DI**stantly **S**upervised en**T**ity sp**AN** de**T**ection and classification). The former entity span detector extracts possible entity mention spans by the distant supervision. Then the later entity classifier assigns each entity span to one of the positive entity types or none by employing a positive and unlabeled (PU) learning framework. Two models were built based on the pre-trained SciBERT model and fine-tuned with the silver corpus generated by the distant supervision. Experimental results on BC5CDR and NCBI-Disease datasets show that our method outperforms the end-to-end NER baselines without PU learning by a large margin. In particular, it increases the recall score effectively.

1 Introduction

The development of Named Entity Recognition (NER) is often hindered by a lack of annotated datasets. For example, medical and biological tasks often differ in target entity types and their granularity levels. In these scenarios, a domain-specific dictionary is often used for distant supervision to search for possible mention spans for the target entity types.

So far, many distantly-supervised NER methods have been proposed. However, the performance of these methods often largely depends on the quality of the domain dictionary. Therefore, it usually necessitates tuning the matching threshold values to account for such noisy and low-coverage labels to obtain optimal results. In this setting, existing end-to-end methods are often not flexible enough to change their internal behaviors.

Hence, we propose a two-step pipeline framework for this task to provide more control knobs to adjust its internal behaviors regarding the quality of the available domain dictionary at hand.

Our method extends span-based NER methods to unsupervised ones (Sohrab and Miwa, 2018; Yongming et al., 2022; Yu et al., 2022); which divides the unsupervised NER task into two consecutive tasks: entity span detection and entity classification. The former extracts textual spans for the candidates of entities from a sentence disregarding the types of entities, then the latter classifies whether it belongs to any predefined entity type or none of them.

Distantly supervised NER often faces a low recall score problem because of the noisy and low-coverage domain dictionary. In addition, entity names in the biological and medical domains have a high cardinality because of synonymous diversity. Hence we employ positive and unlabeled (PU) learning (Liu et al., 2002) in the pipeline to cope with the low recall score problem. Specifically, we define the latter entity classification task as a partially-supervised one by taking the entity spans matched by the dictionary as *positive* and those not matched as *unlabeled* samples. Then using a small portion of the *positive* samples, named *spy*, we probe the behavior of the *unlabeled* samples within the entity classifier. The method is based on the PU learning for binary classification proposed by Liu et al. (2002) and extends it to multinomial classification tasks so that it can be used for NER tasks. The contribution of this work can be summarized as follows:

- We propose a novel distantly-supervised pipeline NER which executes entity span detection and entity classification in sequence by employing a *spy*-based PU learning.
- Experimental results on BC5CDR and NCBI-Disease datasets show 4.8 and 3.2 PP (percentage points) improvement in the F1 scores compared with the best end-to-end NER baseline without PU learning.

The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks, pages 171–177
July 13, 2023 ©2023 Association for Computational Linguistics

2 Related Work

Span-based NER methods (Sohrab and Miwa, 2018; Yongming et al., 2022; Yu et al., 2022) first detect entity spans followed by entity classification, which allows extracting overlapping or nested entities that conventional sequential labeling methods cannot easily handle. Nguyen et al. (2023) employed the information bottleneck principle to enhance span-based NER. Unlike these supervised approaches, our method applied a span-based method to unsupervised NER tasks.

For distantly supervised NER, Shang et al. (2018) proposed AutoNER with a "tie or break" tagging scheme to make the model more amenable to extracting false-negative examples that do not match any dictionary items, coupled with span classification. Liang et al. (2020) proposed BOND, which leverages a pre-trained language model with a self-training. A similar method was proposed by Meng et al. (2021).

Liu et al. (2002) proposed a PU learning method to solve a partially-supervised binary classification by sending known positive spy samples into unknown samples. This allows reliably inferring the behavior of unknown samples in a classification task. Our method extends the method to multinomial classification tasks so that it can be applied to general NER tasks.

PU learning for distantly supervised NER was studied by Peng et al. (2019) based on risk minimization loss defined for binary label classification, which is different from our *spy*-based PU learning.

3 Proposed method

Our task is to extract mention spans that belong to one of predefined entity types $\{e_k|k = 1 \dots K\}$, where K is the number of entity types, from a sentence x by using a domain-specific dictionary $\{d_k|k = 1 \dots K\}$ for each entity type. Optionally, we assume the existence of an auxiliary dictionary d_{aux} without specific entity type information, although we assume that each term of d_{aux} belongs to one of the concepts of $\{e_k\}$.

We define the union of these dictionaries as $d_+ = \{d_k|k = 1 \dots K\} \cup \{d_{aux}\}$.

3.1 Entity spans detection

We employ the SciBERT (Beltagy et al., 2019) tokenizer to tokenize a sentence x into subwords. Let $\{x_i|i = 1..n\}$ be the tokenized sentence of length n. Entity spans are detected by sequential tagging with binary labels $y_i \in \{0, 1\}$. Entity spans are defined as segments of continuous 1s.

To prepare the training dataset, we performed uncased dictionary matching to search for spans of x that match any item of d_+. If multiple spans overlap each other, we take the minimum span that encloses all the overlapping spans.

Let $h_{1:n} = \text{SciBERT}(x_{1:n})$ be the contextualized token embeddings of sentence x from the BERT layers. We transform h_i into a probability $\hat{y}_i \in \mathbb{R}$ by applying a multilayer perceptron (MLP) to h_i with a sigmoid function. We use the following binary cross-entropy L_{span} to train the model;

$$L_{span} = -\frac{1}{n}\sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1-y_i) * \log(1-\hat{y}_i), \quad (1)$$

where y_i is a true span label.

3.2 Entity classification by PU learning

Each entity span detected by the span detector is classified as whether it belongs to any predefined entity type or NA (none of the entity types). Let $\{s^{(j)} = x_{p_j:q_j}|1 \leq p_j < q_j \leq n, j = 1..J\}$, where J is the number of spans detected, be a set of estimated entity spans from p_j-th token to q_j-th token, we label each span with an entity type by dictionary matching to build a silver training corpus for entity classification. We use the snorkel [1] to label each candidate span using the labeling function defined for each type of entity. Although any labeling functions can be used here, we adopt the exact matching to label $\{s^{(j)}\}$ using $\{d_k\}$.

All candidate spans are labeled as positive (belonging to one of $\{e_k\}$) or unlabeled. We use a majority vote to determine the final label of each span. Because we define only one labeling function for each entity type, it is equivalent to a unanimous vote with abstain. Hence a span is matched by none or multiple labeling functions, the label is set to unlabeled. The positively labeled and unlabeled span data are used for PU learning as follows.

The overall framework of PU learning is illustrated in Figure 1, where P and U specify positive and unlabeled span samples, respectively. We assume U contains real negatives belonging to none of the entity types and positives the dictionary matching fails to extract. We want to classify each sample of U, either positive or negative, using P.

[1] https://github.com/snorkel-team/snorkel

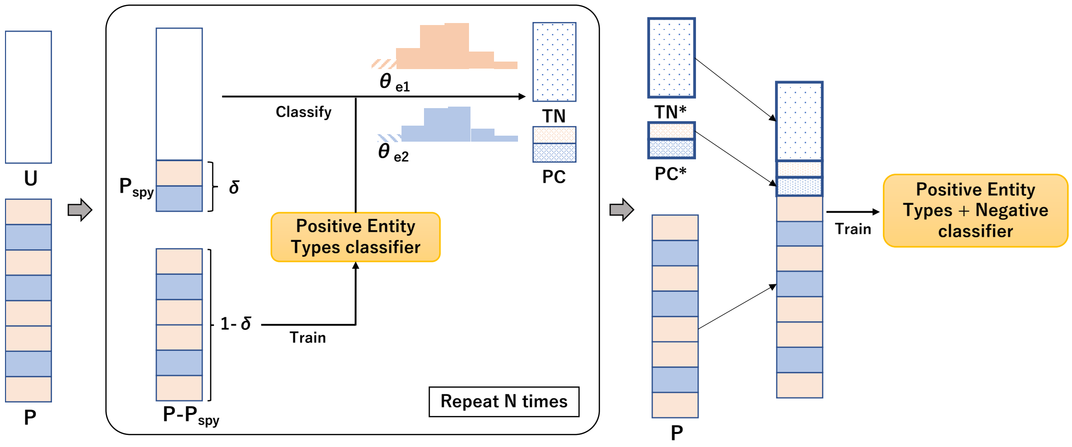

Figure 1: PU learning of an entity classifier. The input is *P* (positive) and *U* (unlabeled) samples. *P* of the two entity types are colored orange and blue. The middle part within the rounded box is repeated N times to generate true negative (*TN*) and positive candidate (*PC*) samples for each round. The intersection of N sets of *TN* and *PC* samples is appended to the original *P*. We use *TN* as negative and *PC*+*P* as positive samples to generate a silver training dataset for the final entity classifier.

To achieve this goal, we use two steps. In the first step, we randomly select a portion δ of P, denoted as P_{spy}, and append them to U. We then train an entity classifier using $P - P_{spy}$ as the training dataset to probe the behavior of each sample of U and P_{spy}. The details are explained below.

We use SciBERT (different from the one employed in span detection) (Beltagy et al., 2019) to get the contextualized embedding of sentence x defined as $h_{1:n}$. For the span $s^{(j)}$, we concatenate the embeddings of the first and last tokens and the average of embeddings corresponding to the span tokens to define the span features defined as follows:

$$f_{span}\left(s^{(j)}\right) = \left[h_{p_j}; \frac{1}{q_j - p_j + 1}\sum_{i=p_j}^{q_j} h_i; h_{q_j}\right], \tag{2}$$

where $[;]$ is a vector concatenation.

Then we employ an MLP with a ReLU function to transform the span features f_{span} to logit vectors of size K. We define a cross-entropy L_{entity} using the predicted class probability $\hat{y}^{(j)} \in \mathbb{R}^K$ corresponding to a span $s^{(j)}$ and the one hot class label $y^{(j)}$ as follows:

$$L_{entity} = \frac{1}{J}\sum_{j=1}^{J} \text{CrossEntropy}\left(\hat{y}^{(j)}, y^{(j)}\right) \tag{3}$$

The trained model classifies U and P_{spy} samples. We use the distribution of k-class probabilities of $\{P_{spy}^{(i)} | s^{(i)} \in e_k\}$, denoted as $\hat{y}_{spy}[k]$, to define a threshold value to judge whether each sample of U is a true negative (TN) or positive candidate (PC).

Specifically, we used the percentile values $\{\theta_k | k = 1 \ldots K\}$ obtained from $\hat{y}_{spy}[k]$ as the threshold value for each type of entity. Assume the maximum class probability of the ith sample of U denoted as $U^{(i)}$ occurs in class k. If its probability is larger than θ_k, the unlabeled sample $U^{(i)}$ is registered as PC of entity class k; otherwise, it is registered as TN. We repeat this PU learning N times by randomly sampling different P_{spy}.

In the second step, we take the intersection of N sets of PC and TN as additional training samples in addition to the original P. We use P+PC as positive and TN as negative samples to train the final entity classifier. We employ the SciBERT-based model as in the first step except for the output dimension, which is $K + 1$ over the total number of positive entity types plus one negative type. During inference, we use only the final entity classifier.

3.3 Details of Training

We used scispacy[2] to tokenize sentences and employed uncased SciBERT as the base model. We used Adam (Kingma and Ba, 2015) optimization with an initial learning rate of $1e - 4$ with a linear decay scheduler with a warmup step of 1,000. We set the batch size of 32 for both the training and validation datasets with early stopping against the validation dataset with maximum patience of 3. We set the maximum epoch size at 50.

[2] https://allenai.github.io/scispacy/

We applied a dropout layer to the input of MLPs with a dropout rate of 0.3. We jointly finetuned the weights of SciBERT and the header layers using L_{span} and L_{entity} for the entity span detector and entity classifier, respectively. We set N of the PU learning to 3.

4 Results

For experiments, we evaluate using BC5CDR (Wei et al., 2016) and NCBI-Disease (Doğan et al., 2014) datasets. BC5CDR dataset contains 1,500 documents with chemical and disease entity annotations. The dataset is split into training, development, and testing data, each with 500 documents; whereas the NCBI-Disease dataset contains 592 training, 100 development, and 100 testing documents with disease annotations. We used only the plain text part of the training and development dataset to train the DISTANT and baseline NERs. We utilized the testing datasets with annotation to evaluate performance.

For the domain dictionaries, we used the dictionary used in AutoNER (Shang et al., 2018) for the BC5CDR dataset, which can be downloaded from the author's GitHub site [3]. The dictionary contains entity items for *Chemical* and *Disease* types and *Others* without entity type information. On the other hand, for the NCBI-Disease dataset, we used the CTD-disease dictionary [4] concatenated with the disease dictionary used for the BC5CDR dataset. Table 3 in Appendix A shows the number of items included for each entity type.

For the baselines, we used naïve dictionary matching, BOND (Liang et al., 2020) and AutoNER (Shang et al., 2018). We also compared our model with the supervised model (SciBERT + CRF) to verify the performance gap between the distantly supervised and the supervised model. All models are implemented by ourselves. We should note that we did not use d_{aux} to train the BOND model because the proposed method does not allow us to use a dictionary without entity types.

Table 1 shows the micro averages of precision, recall, and F1 scores of the NER results of our DISTANT and the baselines for both the BC5CDR and NCBI-Disease datasets. To ignore the performance fluctuations due to the randomness of the DISTANT method, we repeat the experiment three times and average the scores.

For the BC5CDR dataset, since we used preprocessed domain dictionary, the F1 score of the naïve Dictionary match achieved a score of nearly 64%, although AutoNER and our DISTANT both outperformed the naïve method. Compared to the AutoNER model, there was no significant difference in precision. However, our recall score outperformed AutoNER, which resulted in a 3.8 PP increase in the F1 score. Although there is still a relatively large gap between DISTANT and SciBERT+CRF, our method outperformed the baselines.

The second to the last rows show the performance of span detection of DISTANT. Because the number of detectable positive entities is restricted by the result of span detection, the recall score of the span detection indicates the upper bound of the recall score of DISTANT.

For the NCBI-Disease dataset, we had relatively lower performances compared with the results from the BC5CDR dataset. This is because of the poor coverage of the domain dictionary, as indicated by the results of the naïve Dictionary match. Even though the preprocessed disease dictionary was confirmed to cover much of the *Disease* entities in the BC5CDR dataset, the same dictionary failed to capture most of the *Disease* entities in the NCBI-Diesase dataset, even if it was appended by the CTD-disease dictionary.

Although the low performances, the results of AutoNER and DISTANT outperformed the Dictionary match result. We could not acknowledge a large performance gain for BOND against the Dictionary match. Compared with AutoNER, DISTANT improves the recall by more than 10 PP, which resulted in a 3.2 PP increase in the F1 score.

These results indicate that our proposed DISTANT outperformed the baselines in two datasets where we have or do not have a high-quality domain dictionary that covers most of the entities. We should note that the performance of distantly-supervised NER severely depends on the quality of the domain dictionary, and even the naïve dictionary match method with a high-quality domain dictionary would work better than any sophisticated methods if such a dictionary is unavailable.

5 Analysis

Table 2 shows the ablation results when we trained DISTANT without using d_{aux}. The scores in the rows Δ are the comparisons with the original DISTANT scores. From the results, this setting severely

[3] https://github.com/shangjingbo1226/AutoNER
[4] http://ctdbase.org/downloads

	BC5CDR			NCBI-Disease		
Entity types	Chemical, Disease			Disease		
Dictionary	Processed Dict*			Processed Dict* + CTD (disease)		
Method	Precision	Recall	F1(%)	Precision	Recall	F1(%)
SciBERT+CRF†	83.5	86.4	84.9	83.0	85.7	84.3
Dictionary match	74.8	55.7	63.9	31.8	20.1	24.7
BOND	73.8	59.9	66.1	32.1	19.4	24.2
AutoNER	**80.3**	72.2	76.0	**60.1**	23.8	34.1
DISTANT (span detect)	(79.6)	(87.6)	(83.4)	(42.8)	(33.9)	(37.8)
DISTANT	79.7	**81.9**	**80.8**	42.2	**33.5**	**37.3**

Table 1: Results of the baselines and proposed model. SciBERT + CRF† is a supervised model. (*) We used the preprocessed dictionary from MeSH and CTD prepared by Shang et al. (2018).

impacts the recall scores. We speculate that the span detector trained with d_{aux} in addition to $\{d_k\}$ produces more plausible candidate entity spans, which would be treated as false negatives when it is trained without d_{aux}, resulting in much higher recall scores.

	DISTANT*		
	Precision	Recall	F1 (%)
BC5CDR	84.3	62.6	71.5
Δ	+3.7	-19.3	-9.2
NCBI-Disease	48.5	23.6	31.8
Δ	+6.3	-9.9	-5.5

Table 2: Ablation results of DISTANT* without using d_{aux}.

Figure 2 shows the changes in F1 scores of DISTANT evaluated on BC5CDR with respect to three different spy sampling ratios δ (0.05, 0.1, and 0.2) for three different percentile threshold θ (1%, 2.5%, and 5%). As the results illustrate, the F1 scores largely changed from 78.0% to 81.2%, suggesting that the result is sensitive to the threshold values. The optimal combination of δ and θ was obtained as 0.1 and 2.5%. We should note that, in any case, the proposed method is better than AutoNER.

6 Conclusion

We propose a distantly-supervised pipeline NER named DISTANT, which executes entity span detection and entity classification in sequence. We exploit PU learning to train our model. Our model outperformed the end-to-end NER baselines without PU learning by a large margin when the quality of the domain dictionary is relatively high enough. We also confirmed that our method outperformed the dictionary match even if the coverage of the domain dictionary is quite low.

Overall, our method effectively increased recall scores without severely degrading precision scores. The proposed method does not require laborious human annotations and can be applied to any NER task using a domain-specific dictionary.

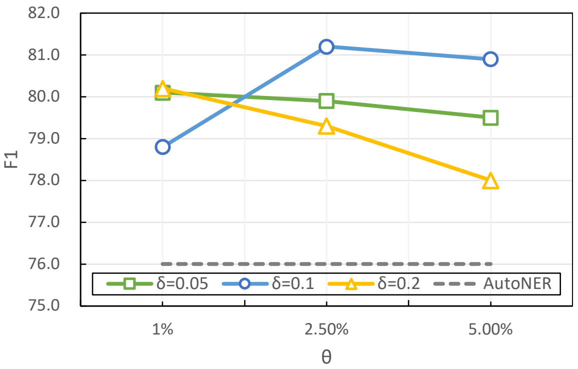

Figure 2: The plot of the F1 scores of DISTANT evaluated on BC5CDR due to the changes in spy samples ratio δ and the percentile threshold θ.

Acknowledgements

This research is supported by the Artificial Intelligence Research Center (AIRC, Japan) and AMED [5] under Grant Numbers JP22ae0101074; JP23ae0101074.

Limitations

The result of Dictionary match and AutoNER based on our implementation is not comparable with

[5] https://www.amed.go.jp/en/

the results shown in the original paper (Shang et al., 2018). Because the performance of distantly-supervised NER is severely dependent on the domain dictionary used, we can not simply compare the performance of the methods if common domain dictionaries are not used.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: A resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics*, 47:1–10.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. 2020. BOND: BERT-Assisted Open-Domain Named Entity Recognition with Distant Supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 1054–1064, New York, NY, USA. Association for Computing Machinery.

Bing Liu, Wee Sun Lee, Philip S. Yu, and Xiaoli Li. 2002. Partially Supervised Classification of Text Documents. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, page 387–394, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Xuan Wang, Yu Zhang, Heng Ji, and Jiawei Han. 2021. Distantly-supervised named entity recognition with noise-robust learning and language model augmented self-training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10367–10378, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nhung T. H. Nguyen, Makoto Miwa, and Sophia Ananiadou. 2023. Span-based named entity recognition by generating and compressing information. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1984–1996, Dubrovnik, Croatia. Association for Computational Linguistics.

Minlong Peng, Xiaoyu Xing, Qi Zhang, Jinlan Fu, and Xuanjing Huang. 2019. Distantly supervised named entity recognition using positive-unlabeled learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2409–2419, Florence, Italy. Association for Computational Linguistics.

Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. 2018. Learning named entity tagger using domain-specific dictionary. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2054–2064, Brussels, Belgium. Association for Computational Linguistics.

Mohammad Golam Sohrab and Makoto Miwa. 2018. Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849, Brussels, Belgium. Association for Computational Linguistics.

Chih-Hsuan Wei, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Jiao Li, Thomas C. Wiegers, and Zhiyong Lu. 2016. Assessing the state of the art in biomedical relation extraction: overview of the BioCreative V chemical-disease relation (CDR) task. *Database*, 2016.

Nian Yongming, Chen Yanping, Qin Yongbin, Huang Ruizhang, Tang Ruixue, and Hu Ying. 2022. A joint model for entity boundary detection and entity span recognition. *Journal of King Saud University - Computer and Information Sciences*, 34(10, Part A):8362–8369.

Jie Yu, Bin Ji, Shasha Li, Jun Ma, Huijun Liu, and Hao Xu. 2022. S-NER: A Concise and Efficient Span-Based Model for Named Entity Recognition. *Sensors*, 22(8).

A Details of domain dictionaries

Table 3 shows the number of items for each entity type included in the domain dictionaries. The BC5CDR dictionary was prepared by Shang et al. (2018); whereas no processing was performed on the CTD (disease) dictionary.

	Processed	Chemical	Disease	Others
BC5CDR*	Yes	1,193	1,289	6,877
CTD (disease)	No		13,261	

Table 3: The number of items for each entity type included in the BC5CDR and CTD (disease) dictionaries. (*) Preprocessed from MeSH and CTD (Shang et al., 2018).